Mars Polar Lander Mission Distributed Operations

Paul G. Backes*, Kam S. Tso†, Jeffrey S. Norris*, Gregory K. Tharp†,
Jeffrey T. Slostad* and Robert G. Bonitz*
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California
†IA Tech Inc., Los Angeles, California

Abstract—The Mars Polar Lander (MPL) mission is the first planetary mission to use Internet-based distributed ground operations where scientists and engineers collaborate in daily mission operations from multiple geographically distributed locations via the Internet. This paper describes the operations system, the Web Interface for Telescience (WITS), which is used by the MPL mission for Internet-based operations. The MPL mission is a lander which will land near the south pole of Mars in December, 1999. WITS will be used for generating command sequences for the lander's stereo camera, robotic arm, and robotic arm camera.

TABLE OF CONTENTS

1. INTRODUCTION

The Mars Polar Lander (MPL) will land near the south pole of Mars in December 1999 and perform an approximately three month mission [1]. An artist's drawing of the lander is shown in Figure 1. The lander will carry the Mars Volatiles and Climate Surveyor (MVACS) instrument suite, lead by scientists at the University of California at Los Angeles (UCLA). The mission operations center will be at UCLA. The mission will include searching for near-surface ice and possible surficial records of cyclic climate change, and characterizing physical processes key to the seasonal cycles of water, carbon dioxide and of dust on Mars.

The MPL mission will be the first NASA planetary mission which utilizes Internet-based ground operations to enable geographically distributed scientists and engineers to collaborate in daily mission command sequence generation. Internet-based distributed operations for the Mars Polar Lander mission will be done using the Web Interface for Telescience (WITS). This paper describes the ground operations for the MPL mission using WITS, the implementation of WITS, and results of operations using WITS.

WITS serves multiple purposes for the MPL mission. It is the primary operations tool for visualization of downlink data and generation of command sequences for the robotic arm (RA) and robotic arm camera (RAC). It is also used for command sequence generation for the Stereo Surface Imager (SSI) stereo camera which is mounted on a lander mast. WITS will also enable Internet-based users to generate command sequences for the RA, RAC, and SSI. For example, scientists at the University of Arizona, who are responsible for the RAC and SSI, will be able to generate daily sequence inputs from Arizona so that they do not have to be at the UCLA operations center for the whole mission. Internet-based operations will also enable WITS support engineers at JPL to provide mission support without having to travel to UCLA. This reduces operations costs by requiring fewer support personnel at UCLA. Also, it enables faster resolution of problems by allowing remote JPL support engineers to work on problems immediately without having to travel to UCLA (an hour or more delay) to address problems. Public outreach is another important application of WITS for the MPL mission. A separate WITS system will be available to the general public to enable them to plan and simulate their own missions.

WITS was originally developed in NASA research programs for use in future rover missions [2], [3]. WITS has been reimplemented based upon rover field test experiences and mission requirements for use in flight missions. The Mars Polar Lander mission is the first flight mission use of WITS. Other examples of Internet-based robot operation can be found in [4], [5].

2. SYSTEM ARCHITECTURE

WITS is a part of the complete MPL mission ground operations system. A simplified diagram of the MPL ground operations system is shown in Figure 2. Downlink data from Mars is processed and put in databases. One of the databases is the WITS database. Sequence generation begins using a sequencing tool called APGEN which generates daily high level sequences for all the lander instruments. The sequences for the different instruments are sent to sequence generation systems specific to each instrument. Included in the sequences are requests which include textural descriptions of what needs to be done in each request and how much time, energy, and data volume is allocated for each request. WITS is the sequence generation system for the RA and RAC and some SSI operations. WITS generates the low-level commands to achieve the goals specified in the requests and within the resource allocations specified. The multiple sequencing systems then output their sequences to the SEQGEN planning tool where all the detailed sequences are integrated and resource checking on the integrated sequence is done and final sequence modifications are made to ensure a valid sequence within resource constraints. The final sequence is then sent into the uplink

[1] 0-7803-5846-5/00/$10.00 © 2000 IEEE

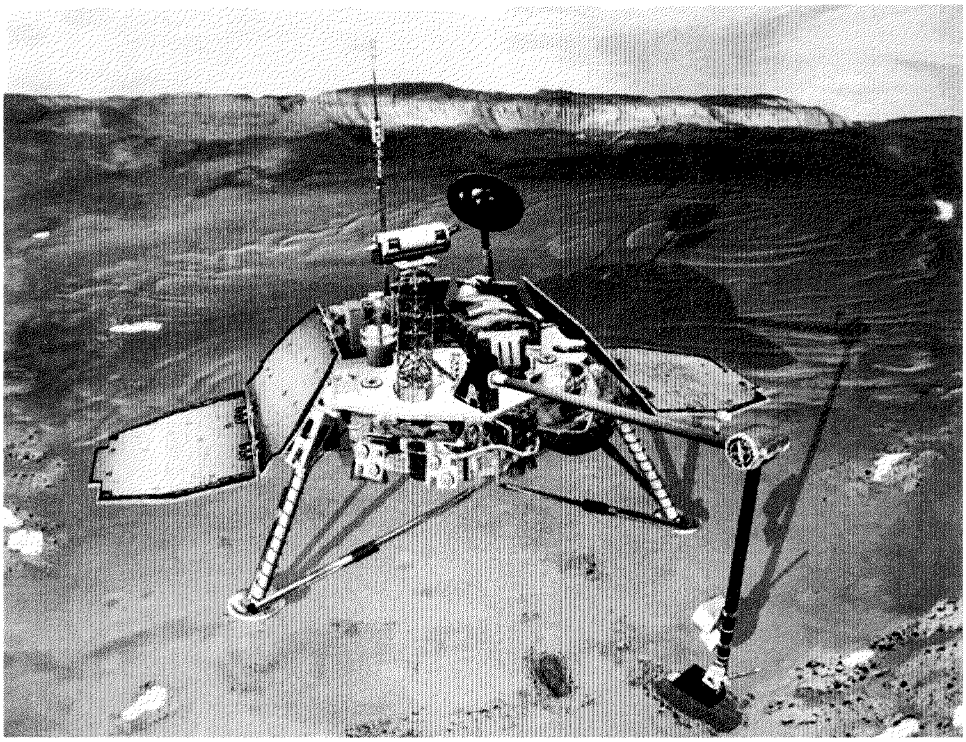


Figure 1. Artist's Drawing of Mars Polar Lander on Mars

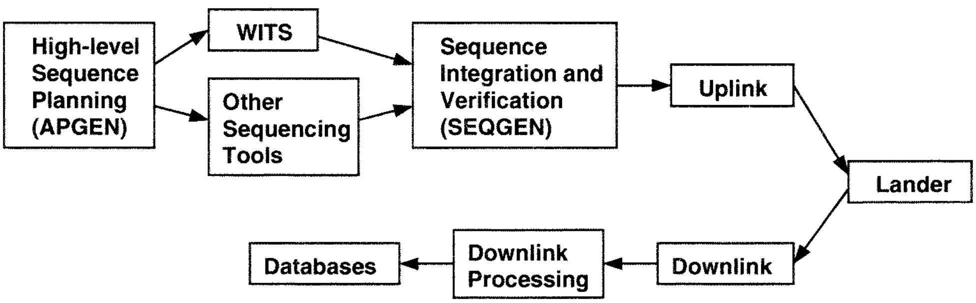


Figure 2. MPL Mission Operations Architecture

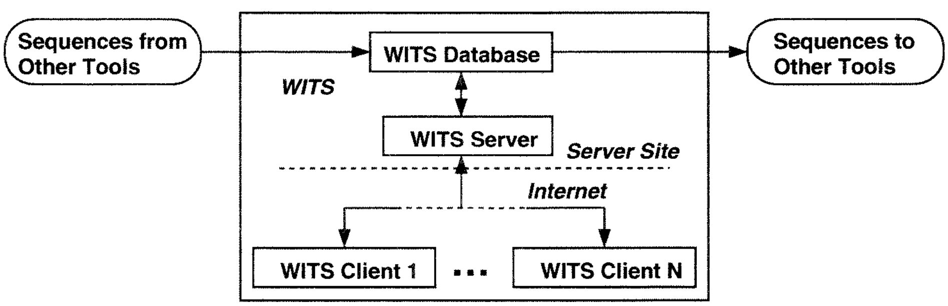


Figure 3. WITS Architecture

process, eventually to be received at the lander.

The WITS architecture is shown in Figure 3. The WITS Database holds downlink data products and uplink sequence information. The WITS Server provides communication between the WITS Database and the WITS Clients. The WITS Clients are distributed over the Internet and provide the interface to the user to view downlink data and generate command sequences. Sequences from other sequencing tools, e.g., APGEN, are placed in the WITS Database for access by the WITS Clients, and sequences completed by WITS are placed in the WITS Database where they are retrieved by other sequencing tools, e.g., SEQGEN. It would be possible to have other sequencing tools communicate with the WITS Server to send and receive sequences between the sequencing tools, but for this mission it was determined that the most efficient approach is to place sequences in the database for other tools to copy.

3. Internet Security

A critical element in Internet-based mission operations is Internet security. Each day, a large amount of data is received from the spacecraft at the mission operations center and placed into a local database for processing and viewing by mission scientists. To enable collaboration in daily sequence generation by distant, Internet-based, scientists, a secure and efficient way to deliver the data to the remote scientists is needed. Individual scientists may want, or have authorization for, only a specific subset of the downlink data. Also, secure Internet communication is needed to receive inputs, e.g., command sequences, from the Internet-based users. WEDDS, the WITS Encrypted Data Delivery System, was created to provide the required secure Internet-based communication for WITS.

WEDDS is a framework for automatically distributing mission data and receiving remote user transmissions over the Internet in a secure and efficient manner. WEDDS has been integrated with WITS for the 1998 Mars Polar Lander mission, but is designed to work with any existing mission application with little modification. WEDDS operates in a fashion that is transparent to the remote user. Files simply appear on the remote user's machine as they become available, and connections are made securely without any additional effort on the part of the user.

WEDDS utilizes a comprehensive approach to Internet security to confirm the identity of its users and encrypt transmissions to and from the mission control center. All WEDDS connections are authenticated using the NASA Public Key Infrastructure (PKI). NASA Ames is currently overseeing the installation of the NASA PKI at all NASA centers [6]. After authentication, all WEDDS communications are made through SSL (Secure Sockets Layer) sockets and are encrypted using the Triple-DES-EDE3 algorithm [7], [8]. Data transferred by WEDDS is nearly impossible to decipher if intercepted, and fooling the authentication protocol would require compromising the NASA Ames Certificate Authority, which is highly protected and forms the backbone of NASA's future approach to security. Each copy of the WEDDS client software can not be activated unless the remote user provides his personal security profile to the system, which is kept on a floppy disk in the possession of the user. Further details on the WEDDS system can be found in [9].

4. Downlink Data Visualization

Downlink data from the lander is provided to the user via various views. Two windows provide the user with available data to be visualized. The Results Tree window displays the available downlink data for the mission by date of downlink. It also has lists of descent imagery. The Plan window displays available Panorama, Overhead, and 3D views for a specific plan. This is usually the most convenient way for a user to specify a view to be opened since the desired views usually have data which are organized for the specific plan. Each of these specific views has a specific set of downlink data it uses, so definitions of these views may be updated each day and put in the new plan. The Results Tree and Plan windows have data displayed in a tree structure whose nodes can be expanded or collapsed. The user opens a view to visualize downlink data by clicking on the item. The various types of views are described below.

The Descent view provides images taken from the spacecraft during descent to the surface and shows the landing location. The Overhead view shows the area around the lander from above. An optional grid provides angle and range information. Color-coded elevation map and texture mapped image options are provided. Targets are displayed in the Overhead view, as well as selected points.

The Panorama view, shown in Figure 4, is a mosaic of images taken by a stereo camera. Selecting a point in an image causes the x,y,z position on the surface to be displayed as well as the ARZ (azimuth angle, range from site center, and z elevation value). The point can be turned into a science target via the Main window Action pull-down menu. The Panorama view can be shown in 1/4, 1/2, and full scale.

The Wedge view displays one image with various options, e.g., left or right image from stereo image pair, anaglyph, and resize. When a user selects a pixel in a Wedge or Panorama image, WITS determines the corresponding x,y,z position and surface normal on the terrain and displays the x,y,z position to the user in all views in which this point is visible. This point can be turned into a target, as described below in Section 5. In the Wedge view, the pixel intensity is also displayed.

The Contrast Adjuster view (selected from a Wedge view pull-down menu) enables the contrast to be adjusted for a Wedge view image. The minimum and maximum desired pixel intensities are selected via scroll bars and then the pixel intensity values of the image are linearly stretched to have the selected pixel intensities become minimum (0) and maximum (255). The histogram of the initial and adjusted images are also shown.

The 3D view, shown in Figure 4, provides a 3D solid model visualization of the lander and terrain. Sequence simulation is visualized in the 3D view.

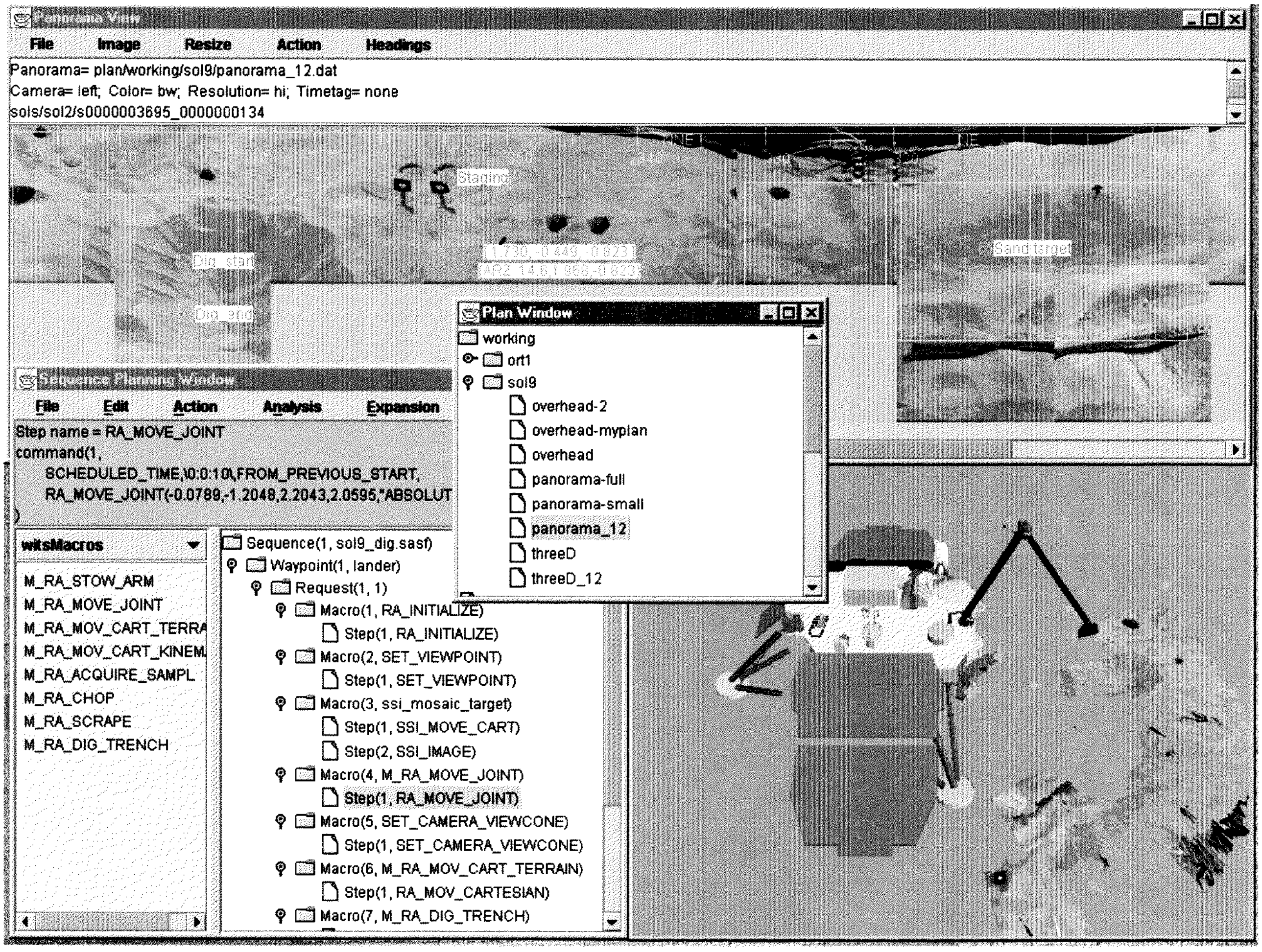


Figure 4. Panorama and 3D Views and Sequence and Plan Windows

5. Sequence Generation

The views discussed above provide a means for visualizing downlink mission data. WITS also provides various windows and features for command sequence generation. WITS enables 3D locations to be used as parameters in commands. As described in the Wedge view description above, a user specifies a 3D point by selecting a pixel in an image in either the Wedge or Panorama views and WITS determines the 3D coordinates and displays them at the selected point. The point can be turned into a target by selecting "Add Target" in the Main window Action pull-down menu. Targets are displayed in the views as pink circles. WITS keeps track of all specified targets and provides them as possible input parameters to sequence commands.

The Sequence window is used to generate a command sequence. A command sequence has a hierarchy of elements. The hierarchy, in descending order, is: Sequence, Waypoint, Request, Macro, Step. There can be any number of elements at a lower level of the hierarchy, e.g., there can be any number of macros in a request. There is only one waypoint for a lander mission, the landing site (the waypoint level is added to enable WITS to support future rover missions as well). A request represents a high-level task. A macro, described in more detail below, is the functional element in WITS by which the user specifies commands and parameters. Macros have expansions into steps. A step is a low-level command that will be uplinked to the spacecraft. WITS can generate various format output sequences. For the flight mission, WITS outputs sequences in the Spacecraft Activity Sequence File (SASF) format [10].

The Sequence window shows the sequences in one plan. Multiple sequences can be displayed. A plan generally represents the planning elements to generate one command sequence to be uplinked to the lander. The sequences are shown on the right hand side of the Sequence window. Supporting multiple sequences is useful for integration of subsequences from different scientists or subsequences for different instruments into the final uplink sequence.

A list of macros which can be inserted into a sequence is shown on the left side of the Sequence window. Multiple lists of macros are available; choosing between macro lists is done via the pull-down menu above the macro list. A macro is inserted into a sequence by selecting the location in the sequence for it to be inserted and then double clicking on the macro in the macro list. Double clicking on a macro in the sequence causes the Macro window to pop up. The structure of the macro window is the same for all macros. At the top is a description area. The parameter names, values, and means for specifying their values is in the middle, and the macro expansion into steps is at the bottom. Depending on the type of parameter, text areas, pull-down menus, and slidebars are provided for specifying the parameter values. A macro-specific algorithm converts the parameters into the expansion which can have any number of steps.

A macro can generate View Objects which are displayed in the views to indicate what the macro is specifying. Figure 4 shows square outlines which are view objects for SSI imaging commands. They represent where images will be taken by the SSI in the current sequence.

There are various sequence editing features in the Sequence window, e.g., cut, copy, paste, and delete in the Action pull-down menu. Additionally, the user can click and drag an item in the sequence to another position in the sequence, e.g., the user can click on a macro and drag it into a different request.

The WITS Sequence Execution window allows the user to simulate a sequence. The sequence simulation is visualized in the 3D view.

Resource analysis and rules checking are important elements of sequence generation. WITS provides resource analysis for a sequence. The duration, energy, and data volume for each step of the sequence are computed and stored along with the cumulative duration, energy and data volume at each step. When a user clicks on a step in the Sequence Window, then information about that step is displayed in the text area at the top of the window, e.g., resources for that step and cumulative resources through that step and absolute execution time. Rules checking is important to ensure that a sequence is valid relative to specified sequence rules. An example of MPL mission sequence rules that WITS checks is verification that request resources used are within allocations.

6. Distributed Collaboration

Internet-based collaboration is achieved in WITS by providing Internet-based users with daily downlink data and allowing them to specify targets and generate command sequences and save them to the common server. The Internet-based users can see each others targets and sequences and can use inputs from each-other's sequences. The Sequence window File pull-down menu enables sequences to be loaded from the common server (e.g., located at UCLA), or to be saved to the common server. One of the uses of distributed operations for the MPL mission is to to have scientists at the University of Arizona generate SSI imaging sequences and submit them to the UCLA operations site.

7. Public Outreach

An important motivation for the development and use of WITS in a planetary mission is its use in public outreach. During the mission a separate version of WITS will be made available to the general public to run from any location on the Internet, e.g., from their homes. A subset of mission downlink data will be put in the public WITS database. The public will then be able to view mission downlink data and plan and simulate their own missions. The goal is to provide the public with an engaging mission experience where they use the same tool as the mission scientists to view mission data and generate and simulate their own missions. The URL that the public will go to from their browser is http://robotics.jpl.nasa.gov/tasks/wits/. The public outreach site will be sponsored by Sun Microsystems at Graham Technology Solutions, Inc.

8. Implementation

The client WITS system which is used by the Internet-based users was implemented using the Java1.2 programming language. The system is accessed by remote users either by using the Java1.2 appletviewer or by going to a URL from a web browser. At this time, only browsers running on Windows95/98/NT operating systems can access WITS via a browser since Java1.2 is not yet supported for browsers on other operating systems. The two browsers which have been used are Netscape Navigator and Microsoft Internet Explorer. In this case, the user first must download the Java PlugIn in order for the browser to support Java1.2 language applets. Users who have computers with Sun Solaris operating systems download the Java Development Kit (JDK) onto their computers and access and run WITS from a URL using the JDK appletviewer application. It is anticipated that Netscape Navigator will soon support Java1.2 for the Solaris operating system, so those users will be able to access WITS via their browser as well. Java3D must be loaded onto the client computers in order to see the 3D view. The database is a structured Unix file system.

9. Mission Results

WITS is currently being used in pre-landing operations tests at the UCLA operations center. The final version of this paper will include results of the mission through the final submission deadline.

10. Conclusions

WITS is providing a new Internet-based operations paradigm for planetary mission operations. WITS is being used in the Mars Polar Lander mission ground operations for downlink data visualization and command sequence generation for the robotic arm, robotic arm camera, and stereo surface imager. With the use of WITS, the MPL mission is the first planetary mission to utilize Internet-based ground operations. The integrated visualization, sequence planning, and Internet security features of WITS make it a valuable tool for planetary mission operations.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] NASA/UCLA. *Mars Surveyor 98 Lander*, 1999. http://mars.jpl.nasa.gov/msp98/lander/.

[2] Paul G. Backes, Gregory K. Tharp, and Kam S. Tso. The web interface for telescience (WITS). In *Proceedings IEEE International Conference on Robotics and Automation*, pages 411–417, Albuquerque, New Mexico, April 1997.

[3] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, and R. Petras. The Rocky7 rover: A Mars sciencecraft prototype. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2458–2464, Albuquerque, New Mexico, April 1997.

[4] Eric Paulos and John Canny. Delivering real reality to the World Wide Web via telerobotics. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1694–1699, Minneapolis, Minnesota, April 22-28 1996.

[5] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Robot teleoperation via WWW. In *Proceedings IEEE International Conference on Robotics and Automation*, May 1995.

[6] NASA-Ames. *NASA Ames IT Security Development Group*, 1999. http://pki.arc.nasa.gov/.

[7] Netscape. *SSL 3.0 Specification*, 1999. http://home.netscape.com/eng/ssl3/.

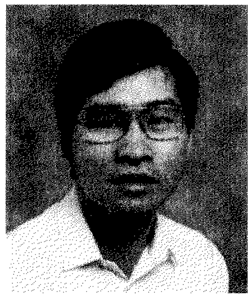
[8] RSA Data Security, Inc. *RSA Labs FAQ- What is Triple-DES?*, 1999. http://www.rsa.com/rsalabs/faq/html/3-2-6.html.

[9] Jeffrey S. Norris and Paul G. Backes. Wedds: The wits encrypted data delivery system. In *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 2000.

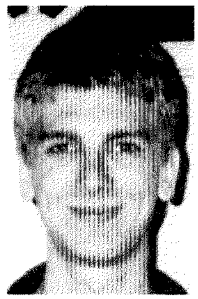
[10] W. Lombard. Spacecraft activity sequence file. Technical Report JPL Software Interface Specification Module (SFOC-1-SEQ-ANY-SASF), Revision 06, Jet Propulsion Laboratory, 1997.

BIOGRAPHY

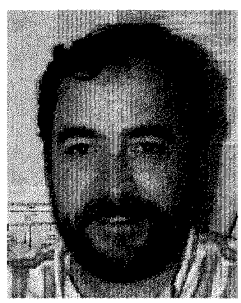
Paul Backes *is a technical group leader in the Autonomy and Control section at Jet Propulsion Laboratory, Pasadena, CA, where he has been since 1987. He received the BSME degree from U.C. Berkeley in 1982, and MSME in 1984 and Ph.D. in 1987 in Mechanical Engineering from Purdue University. He is currently responsible for distributed operations research for Mars lander and rover missions at JPL. Dr. Backes received the 1993 NASA Exceptional Engineering Achievement Medal for his contributions to space telerobotics (one of thirteen throughout NASA), 1993 Space Station Award of Merit, Best Paper Award at the 1994 World Automation Congress, 1995 JPL Technology and Applications Program Exceptional Service Award, 1998 JPL Award for Excellence and 1998 Sole Runner-up NASA Software of the Year Award. He has served as an Associate Editor of the IEEE Robotics and Automation Society Magazine.*



Kam Sing Tso *received the B.S. degree in Electronics from the Chinese University of Hong Kong, Hong Kong, in 1979, M.S. degree in Electronic Engineering from the Philips International Institute, Eindhoven, the Netherlands, in 1981, and Ph.D. degree in Computer Science from the University of California, Los Angeles, in 1987. From 1986 to 1996, he worked at the Jet Propulsion Laboratory and SoHaR Inc. conducting research and development on robotics systems, fault-tolerant systems, and reliable software. He is currently the Vice President of IA Tech, Inc. Dr. Tso's research interests include World Wide Web technologies, distributed simulation and collaboration, high performance and dependable real-time software and systems.*



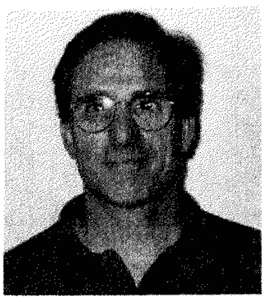
Jeffrey S. Norris *is a computer scientist and member of the technical staff of the Autonomy and Control Section at the Jet Propulsion Laboratory. He specializes in software engineering for telerobotics, distributed operations, machine vision, and large scale interfaces. He received his Bachelor's and Master's degrees in Electrical Engineering and Computer Science from MIT. While an undergraduate, he worked at the MIT Media Laboratory on data visualization and media transport protocols. He completed his Master's thesis on face detection and recognition at the MIT Artificial Intelligence Laboratory. He now lives with his wife in Azusa, California.*



Gregory K. Tharp *received the BSME degree from the University of California, Berkeley, holds the B.A. in economics from the University of California, Santa Cruz, and received* *the MSME degree from the University of California, Berkeley, in 1989. After a one year staff appointment at the University of California, Berkeley, he worked as a consultant for Fujita Research for three years, studying virtual environment and telepresent display systems for remote manipulation. From 1993 until 1997 he was a member of technical staff at the Jet Propulsion Laboratory, California Institute of Technology working on graphical user interfaces and image processing for the NASA robotics program. Currently, he is a research engineer at IA Tech Inc. where he is developing a collaborative distributed environment for planning and control of remote autonomous systems.*

Jeff Slostad picture will be placed here

Jeffrey T. Slostad *is a member of the technical staff in the Autonomy and Control section at Jet Propulsion Laboratory, Pasadena, CA. He is currently the Robotic Arm Team Lead for the Mars Polar Lander mission.*



Robert G. Bonitz *Received the Ph.D. degree from the University of California at Davis. He is a member of the technical staff in the Autonomy and Control section at Jet Propulsion Laboratory, Pasadena, CA. He is currently the lead for robotic arm controls in the Mars Polar Lander mission.*